Information Visualization and Visual Data Mining

Daniel A. Keim, *Member*, *IEEE Computer Society*

Abstract—Never before in history has data been generated at such high volumes as it is today. Exploring and analyzing the vast volumes of data is becoming increasingly difficult. Information visualization and visual data mining can help to deal with the flood of information. The advantage of visual data exploration is that the user is directly involved in the data mining process. There are a large number of information visualization techniques which have been developed over the last decade to support the exploration of large data sets. In this paper, we propose a classification of information visualization and visual data mining techniques which is based on the *data type to be visualized*, the *visualization technique*, and the *interaction and distortion technique*. We exemplify the classification using a few examples, most of them referring to techniques and systems presented in this special section.

Index Terms—Information visualization, visual data mining, visual data exploration, classification.

1 Introduction

The progress made in hardware technology allows today's computer systems to store very large amounts of data. Researchers from the University of Berkeley estimate that, every year, about 1 Exabyte (= 1 Million Terabytes) of data are generated, of which a large portion is available in digital form. This means that, in the next three years, more data will be generated than in all of human history before. The data is often automatically recorded via sensors and monitoring systems. Even simple transactions of everyday life, such as paying by credit card or using the telephone, are typically recorded by computers. Usually, many parameters are recorded, resulting in multidimensional data with a high dimensionality. The data of all mentioned areas is collected because people believe that it is a potential source of valuable information, providing a competitive advantage (at some point). Finding the valuable information hidden in them, however, is a difficult task. With today's data management systems, it is only possible to view quite small portions of the data. If the data is presented textually, the amount of data which can be displayed is in the range of some 100 data items, but this is like a drop in the ocean when dealing with data sets containing millions of data items. Having no possibility of adequately exploring the large amounts of data which have been collected because of their potential usefulness, the data becomes useless and the databases become data "dumps."

1.1 Benefits of Visual Data Exploration

For data mining to be effective, it is important to include the human in the data exploration process and combine the flexibility, creativity, and general knowledge of the human with the enormous storage capacity and the computational power of today's computers. Visual data exploration aims at integrating the human in the data exploration process, applying its perceptual abilities to the large data sets available in today's computer systems. The basic idea of visual data exploration is to present the data in some visual form, allowing the human to get insight into the data, draw conclusions, and directly interact with the data. Visual data mining techniques have proven to be of high value in exploratory data analysis and they also have high potential for exploring large databases. Visual data exploration is especially useful when little is known about the data and the exploration goals are vague. Since the user is directly involved in the exploration process, shifting and adjusting the exploration goals is automatically done if necessary.

The visual data exploration process can be seen as a hypothesis generation process: The visualizations of the data allow the user to gain insight into the data and come up with new hypotheses. The verification of the hypotheses can also be done via visual data exploration, but it may also be accomplished by automatic techniques from statistics or machine learning. In addition to the direct involvement of the user, the main advantages of visual data exploration over automatic data mining techniques from statistics or machine learning are:

- Visual data exploration can easily deal with highly nonhomogeneous and noisy data,
- Visual data exploration is intuitive and requires no understanding of complex mathematical or statistical algorithms or parameters.

As a result, visual data exploration usually allows a faster data exploration and often provides better results, especially in cases where automatic algorithms fail. In addition, visual data exploration techniques provide a much higher degree of confidence in the findings of the exploration. This fact leads to a high demand for visual exploration techniques and makes them indispensable in conjunction with automatic exploration techniques.

- *The author is with AT&T Shannon Research Labs, Florham Park, NJ and the University of Constance, Germany.*
  E-mail: keim@research.att.com.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number 114974.

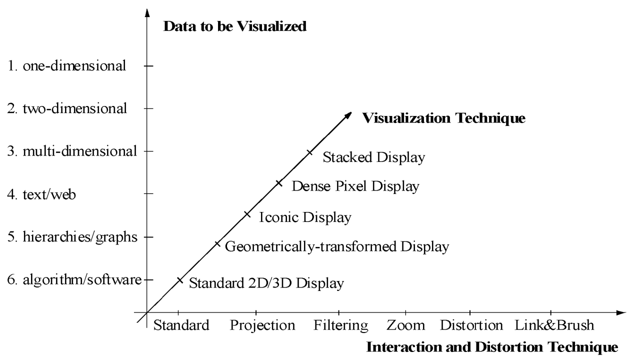

Fig. 1. Classification of information visualization techniques.

1.2 Visual Exploration Paradigm

Visual Data Exploration usually follows a three step process: *Overview first, zoom and filter, and then details-on-demand* (which has been called the Information Seeking Mantra [1]). First, the user needs to get an overview of the data. In the overview, the user identifies interesting patterns and focuses on one or more of them. For analyzing the patterns, the user needs to drill down and access details of the data. Visualization technology may be used for all three steps of the data exploration process: Visualization techniques are useful for showing an overview of the data, allowing the user to identify interesting subsets. In this step, it is important to keep the overview visualization while focusing on the subset using another visualization technique. An alternative is to distort the overview visualization in order to focus on the interesting subsets. To further explore the interesting subsets, the user needs a drill-down capability in order to get the details about the data. Note that visualization technology not only provides the base visualization techniques for all three steps, but also bridges the gaps between the steps.

2 Classification of Visual Data Mining Techniques

Information visualization focuses on data sets lacking inherent 2D or 3D semantics and therefore also lacking a standard mapping of the abstract data onto the physical screen space. There are a number of well-known techniques for visualizing such data sets, such as x-y plots, line plots, and histograms. These techniques are useful for data exploration, but are limited to relatively small and low-dimensional data sets. In the last decade, a large number of novel information visualization techniques have been developed, allowing visualizations of multidimensional data sets without inherent two or three-dimensional semantics. Nice overviews of the approaches can be found in a number of recent books [2], [3], [4], [5]. The techniques can be classified based on three criteria (see Fig. 1) [6]: the data to be visualized, the visualization technique, and the interaction and distortion technique used.

The **data type to be visualized** [1] may be

- one-dimensional data, such as temporal data as used in ThemeRiver (see Fig. 2 in [7]),
- two-dimensional data, such as geographical maps as used in Polaris (see Fig. 3(c) in [8]) and MGV (see Fig. 9 in [9]),
- multidimensional data, such as relational tables as used in Polaris (see Fig. 6 in [8]) and the Scalable Framework (see Fig. 1 in [10]),
- text and hypertext, such as news articles and Web documents as used in ThemeRiver (see Fig. 2 in [7]),
- hierarchies and graphs, such as telephone calls and Web documents as used in MGV (see Fig. 13 in [9]) and the Scalable Framework (see Fig. 7 in [10]),
- algorithms and software, such as the debugging operations as used in Polaris (see Fig. 7 in [8]).

The **visualization technique** used may be classified into

- standard 2D/3D displays, such as bar charts and x-y plots, as used in Polaris (see Fig. 1 in [8]),
- geometrically transformed displays, such as landscapes and parallel coordinates, as used in Scalable Framework (see Figs. 2 and 12 in [10]),
- icon-based displays, such as needle icons and star icons, as used in MGV (see Figs. 5 and 6 in [9]),
- dense pixel displays, such as the recursive pattern and circle segments techniques (see Figs. 3 and 4) [11] and the graph scetches as used in MGV (see Fig. 4 in [9]),
- stacked displays, such as treemaps [12], [13] or dimensional stacking [14].

The third dimension of the classification is the **interaction and distortion technique** used. Interaction and distortion techniques allow users to directly interact with the visualizations. They may be classified into:

- Interactive Projection, as used in the GrandTour system [15],
- Interactive Filtering, as used in Polaris (see Fig. 6 in [8]),
- Interactive Zooming, as used in MGV and the Scalable Framework (see Fig. 8 in [10]),
- Interactive Distortion, as used in the Scalable Framework (see Fig. 7 in [10]),
- Interactive Linking and Brushing, as used in Polaris (see Fig. 7 in [8]) and the Scalable Framework (see Figs. 12 and 14 in [10]).

Note that the three dimensions of our classification—data type to be visualized, visualization technique, and interaction and distortion technique—can be assumed to be orthogonal. Orthogonality means that any of the visualization techniques may be used in conjunction with any of the interaction techniques as well as any of the distortion techniques for any data type. Note also that a specific system may be designed to support different data types and that it may use a combination of multiple visualization and interaction techniques.

3 Data Type to Be Visualized

In information visualization, the data usually consists of a large number of records, each consisting of a number of variables or dimensions. Each record corresponds to an observation, measurement, transaction, etc. Examples are

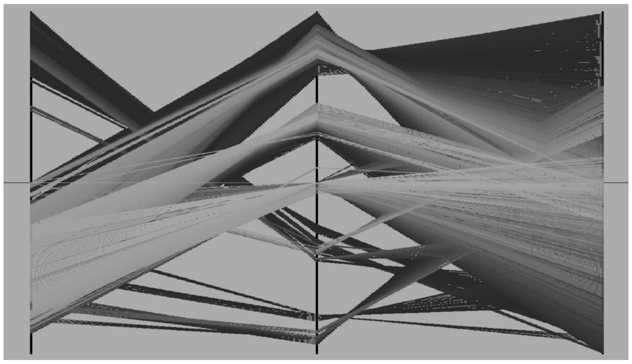

Fig. 2. Parallel coordinate visualization ©IEEE.

customer properties, e-commerce transactions, and physical experiments. The number of attributes can differ from data set to data set: One particular physical experiment, for example, can be described by five variables, while another may need hundreds of variables. We call the number of variables the dimensionality of the data set. Data sets may be one-dimensional, two-dimensional, multidimensional, or may have more complex data types, such as text/hypertext or hierarchies/graphs. Sometimes, a distinction is made between dense (or grid) dimensions and the dimensions which may have arbitrary values. Depending on the number of dimensions with arbitrary values, the data is sometimes also called univariate, bivariate, etc.

3.1 One-Dimensional Data

One-dimensional data usually has one dense dimension. A typical example of one-dimensional data is temporal data. Note that, with each point of time, one or multiple data values may be associated. Examples are time series of stock prices (see Fig. 3 and Fig. 4 for an example) or the time series of news data used in the ThemeRiver examples (see Figs. 2-5 in [7]).

3.2 Two-Dimensional Data

Two-dimensional data has two distinct dimensions. A typical example is geographical data, where the two distinct dimensions are longitude and latitude. X-Y-plots are a typical method for showing two-dimensional data and maps are a special type of x-y-plots for showing two-dimensional geographical data. Examples are the geographical maps used in Polaris (see Fig. 3(c) in [8]) and in MGV (see Fig. 9 in [9]). Although it seems easy to deal with temporal or geographic data, caution is advised. If the number of records to be visualized is large, temporal axes and maps quickly get glutted—and may not help to understand the data.

3.3 Multidimensional Data

Many data sets consists of more than three attributes and, therefore, they do not allow a simple visualization as two-dimensional or three-dimensional plots. Examples of multidimensional (or multivariate) data are tables from relational databases, which often have tens to hundreds of columns (or attributes). Since there is no simple mapping of the attributes to the two dimensions of the screen, more sophisticated visualization techniques are needed. An example of a technique which allows the visualization of multidimensional data is the Parallel Coordinate Technique [16] (see Fig. 2, which is also used in the Scalable Framework (see Fig. 12 in [10]). Parallel Coordinates display each multidimensional data item as a polygonal line which intersects the horizontal dimension axes at the position corresponding to the data value for the corresponding dimension.

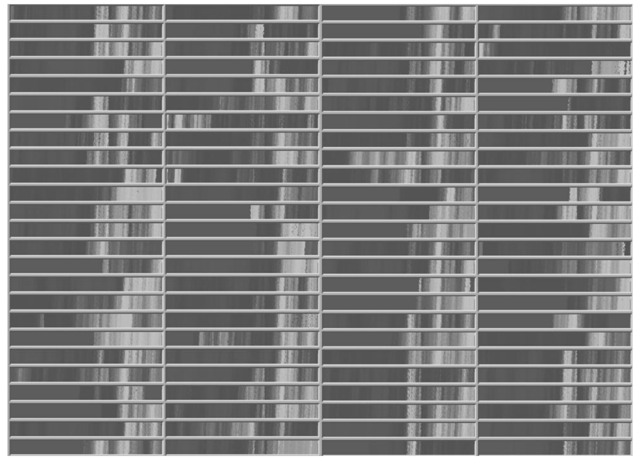

Fig. 3. Dense pixel displays: recursive pattern technique © IEEE.

3.4 Text and Hypertext

Not all data types can be described in terms of dimensionality. In the age of the world wide web, one important data type is text and hypertext as well as multimedia web page contents. These data types differ in that they cannot be easily described by numbers and, therefore, most of the standard visualization techniques cannot be applied. In most cases, a transformation of the data into description vectors is necessary first before visualization techniques can be used. An example for a simple transformation is word counting (see ThemeRiver [7]), which is often combined with a principal component analysis or multidimensional scaling (for example, see [17]).

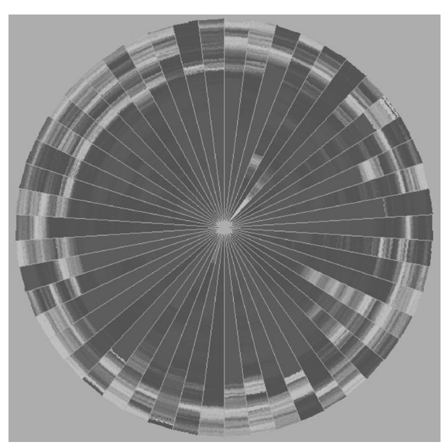

Fig. 4. Dense pixel displays: circle segments technique © IEEE.

3.5 Hierarchies and Graphs

Data records often have some relationship to other pieces of information. Graphs are widely used to represent such interdependencies. A graph consists of a set of objects, called nodes, and connections between these objects, called edges. Examples are the e-mail interrelationships among people, their shopping behavior, the file structure of the hard disk, or the hyperlinks in the world wide web. There are a number of specific visualization techniques that deal with hierarchical and graphical data. A nice overview of hierachical information visualization techniques can be found in [18], an overview of web visualization techniques at [19], and an overview book on all aspects related to graph drawing is [20].

3.6 Algorithms and Software

Another class of data are algorithms and software. Coping with large software projects is a challenge. The goal of visualization is to support software development by helping to understand algorithms, e.g., by showing the flow of information in a program, to enhance the understanding of written code, e.g., by representing the structure of thousands of source code lines as graphs, and to support the programmer in debugging the code, i.e., by visualizing errors. There are a large number of tools and systems which support these tasks. An nice overview can be found in [21].

4 Visualization Techniques

There is a large number of visualization techniques which can be used for visualizing the data. In addition to standard 2D/3D-techniques, such as x-y (x-y-z) plots, bar charts, line graphs, etc., there are a number of more sophisticated visualization techniques. The classes correspond to basic visualization principles which may be combined in order to implement a specific visualization system.

4.1 Geometrically Transformed Displays

Geometrically transformed display techniques aim at finding "interesting" transformations of multidimensional data sets. The class of geometric display techniques includes techniques from exploratory statistics, such as scatterplot matrices [22], [23] and techniques which can be subsumed under the term "projection pursuit" [24]. Other geometric projection techniques include Prosection Views [25], [26], Hyperslice [27], and the well-known Parallel Coordinates visualization technique [16]. The *parallel coordinate technique* maps the k-dimensional space onto the two display dimensions by using k equidistant axes which are parallel to one of the display axes. The axes corespond to the dimensions and are linearly scaled from the minimum to the maximum value of the corresponding dimension. Each data item is presented as a polygonal line, intersecting each of the axes at that point which corresponds to the value of the considered dimensions (see Fig. 2).

4.2 Iconic Displays

Another class of visual data exploration techniques are the iconic display techniques. The idea is to map the attribute values of a multidimensional data item to the features of an icon. Icons can be arbitraily defined: They may be little faces [28], needle icons as used in MGV (see Fig. 5 in [9]), star icons [14], stick figure icons [29], color icons [30], [31], and TileBars [32]. The visualization is generated by mapping the attribute values of each data record to the features of the icons. In the case of the stick figure technique, for example, two dimensions are mapped to the display dimensions and the remaining dimensions are mapped to the angles and/or limb length of the stick figure icon. If the data items are relatively dense with respect to the two display dimensions, the resulting visualization presents texture patterns that vary according to the characteristics of the data and are therefore detectable by preattentive perception.

4.3 Dense Pixel Displays

The basic idea of dense pixel techniques is to map each dimension value to a colored pixel and group the pixels belonging to each dimension into adjacent areas [11]. Since, in general, dense pixel displays use one pixel per data value, the techniques allow the visualization of the largest amount of data possible on current displays (up to about 1,000,000 data values). If each data value is represented by one pixel, the main question is how to arrange the pixels on the screen. Dense pixel techniques use different arrangments for different purposes. By arranging the pixels in an appropriate way, the resulting visualization provides detailed information on local correlations, dependencies, and hot spots.

Well-known examples are the recursive pattern technique [33] and the circle segments technique [34]. The *recursive pattern technique* is based on a generic recursive back-and-forth arrangement of the pixels and is particularly aimed at representing datasets with a natural order according to one attribute (e.g., time series data). The user may specify parameters for each recursion level and thereby control the arrangement of the pixels to form semantically meaningful substructures. The base element on each recursion level is a pattern of height h_i and width w_i as specified by the user. First, the elements correspond to single pixels which are arranged within a rectangle of height h_1 and width w_1 from left to right, then below backward from right to left, then again forward from left to right, and so on. The same basic arrangement is done on all recursion levels with the only difference that the basic elements which are arranged on level i are the pattern resulting from the level $(i-1)$ arrangements. In Fig. 3, an example recursive pattern visualization of financial data is shown. The visualization shows 20 years (January 1974-April 1995) of daily prices of the 100 stocks contained in the Frankfurt Stock Index (FAZ). The idea of the *circle segments technique* [34] is to represent the data in a circle which is divided into segments, one for each attribute. Within the segments, each attribute value is again visualized by a single colored pixel. The arrangment of the pixels starts at the center of the circle and continues to the outside by plotting on a line orthogonal to the segment halving line in a back and forth manner. The rationale of this approach is that, close to the center, all attributes are close to each other, enhancing the visual comparison of their values. Fig. 4 shows an example circle segment visualization of the same data (50 stocks) as shown in Fig. 3.

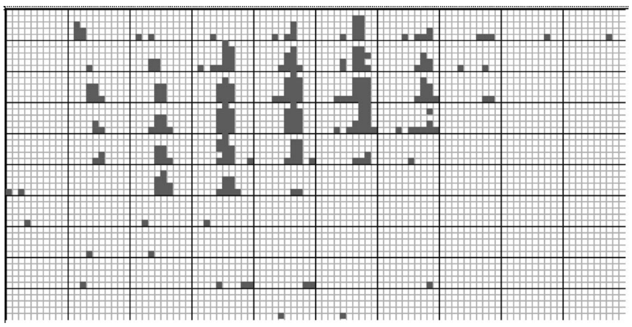

Fig. 5. Dimensional stacking visualization of oil mining data (used by permission of M. Ward, Worchester Polytechnic © IEEE).

4.4 Stacked Displays

Stacked display techniques are tailored to present data partitioned in a hierarchical fashion. In the case of multi-dimensional data, the data dimensions to be used for partitioning the data and building the hierarchy have to be selected appropriately. An example of a stacked display technique is *Dimensional Stacking* [35]. The basic idea is to embed one coordinate system inside another coordinate system, i.e., two attributes form the outer coordinate system, two other attributes are embedded into the outer coordinate system, and so on. The display is generated by dividing the outmost level coordinate systems into rectangular cells and, within the cells, the next two attributes are used to span the second level coordinate system. This process may be repeated one more time. The usefulness of the resulting visualization largely depends on the data distribution of the outer coordinates and, therefore, the dimensions which are used for defining the outer coordinate system have to be selected carefully. A rule of thumb is to choose the most important dimensions first. A dimensional stacking visualization of oil mining data with longitude and latitude mapped to the outer x and y axes, as well as ore grade and depth mapped to the inner x and y axes, is shown in Fig. 5. Other examples of stacked display techniques include Worlds-within-Worlds [36], Treemap [12], [13], and Cone Trees [37].

5 Interaction and Distortion Techniques

In addition to the visualization technique, for an effective data exploration, it is necessary to use some interaction and distortion techniques. *Interaction techniques* allow the data analyst to directly interact with the visualizations and dynamically change the visualizations according to the exploration objectives and they also make it possible to relate and combine multiple independent visualizations. *Distortion techniques* help in the data exploration process by providing means for focusing on details while preserving an overview of the data. The basic idea of distortion techniques is to show portions of the data with a high level of detail, while others are shown with a lower level of detail. We distinguish between the terms *dynamic* and *interactive*, depending on whether the changes to the visualizations are made automatically or manually (by direct user interaction).

5.1 Dynamic Projections

The basic idea of dynamic projections is to dynamically change the projections in order to explore a multidimensional data set. A classic example is the GrandTour system [15], which tries to show all interesting two-dimensional projections of a multidimensional data set as a series of scatter plots. Note that the number of possible projections is exponential in the number of dimensions, i.e., it is intractable for a large dimensionality. The sequence of projections shown can be random, manual, precomputed, or data driven. Systems supporting dynamic projection techniques are XGobi [38], [39], XLispStat [40], and ExplorN [41].

5.2 Interactive Filtering

In exploring large data sets, it is important to interactively partition the data set into segments and focus on interesting subsets. This can be done by a direct selection of the desired subset (*browsing*) or by a specification of properties of the desired subset (*querying*). Browsing is very difficult for very large data sets and querying often does not produce the desired results. Therefore, a number of interaction techniques have been developed to improve interactive filtering in data exploration. An example of an interactive tool which can be used for interactive filtering is Magic Lenses [42], [43]. The basic idea of Magic Lenses is to use a tool like a magnifying glass to support filtering the data directly in the visualization. The data under the magnifying glass is processed by the filter and the result is displayed differently than the remaining data set. Magic Lenses show a modified view of the selected region, while the rest of the visualization remains unaffected. Note that several lenses with different filters may be used; if the filters overlap, all filters are combined. Other examples of interactive filtering techniques and tools are InfoCrystal [44], Dynamic Queries [45], [46], [47], and Polaris [8] (see Fig. 6 in [8] for an example).

5.3 Interactive Zooming

Zooming is a well-known technique which is widely used in a number of applications. In dealing with large amounts of data, it is important to present the data in a highly compressed form to provide an overview of the data, but, at the same time, allow a variable display of the data on different resolutions. Zooming not only means to display the data objects larger, but also means that the data representation automatically changes to present more details on higher zoom levels. The objects may, for example, be represented as single pixels on a low zoom level, as icons on an intermediate zoom level, and as labeled objects on a high resolution. An interesting example applying the zooming idea to large tabular data sets is the TableLens approach [48]. Getting an overview of large tabular data sets is difficult if the data is displayed in textual form. The basic idea of TableLens is to represent each numerical value by a small bar. All bars have a one-pixel height and the lengths are determined by the attribute values. This means that the number of rows on the display can be nearly as high as the vertical resolution and the number of columns depends on the maximum width of the bars for each attribute. The initial view allows the user to detect patterns, correlations, and outliers in the data set. In order to explore

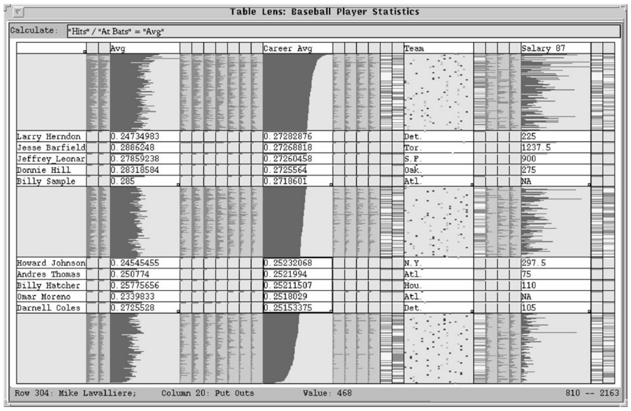

Fig. 6. Table Lenses (used by permission of R. Rao, Xerox PARC © ACM).

a region of interest, the user can zoom in, with the result that the affected rows (or columns) are displayed in more detail, possibly even in textual form. Fig. 6 shows an example of a baseball database with a few rows being selected in full detail. Other examples of techniques and systems which use interactive zooming include PAD++ [49], [50], [51], IVEE/Spotfire [52], and DataSpace [53]. A comparison of fisheye and zooming techniques can be found in [54].

5.4 Interactive Distortion

Interactive distortion techniques support the data exploration process by preserving an overview of the data during drill-down operations. The basic idea is to show portions of the data with a high level of detail while others are shown with a lower level of detail. Popular distortion techniques are hyperbolic and spherical distortions, which are often used on hierarchies or graphs, but may be also applied to any other visualization technique. An example of spherical distortions is provided in the Scalable Framework paper (see Fig. 5 in [10]). An overview of distortion techniques is provided in [55] and [56]. Examples of distortion techniques include Bifocal Displays [57], Perspective Wall [58], Graphical Fisheye Views [59], [60], Hyperbolic Visualization [61], [62], and Hyperbox [63].

5.5 Interactive Linking and Brushing

There are many possibilities to visualize multidimensional data, but all of them have some strengths and some weaknesses. The idea of linking and brushing is to combine different visualization methods to overcome the shortcomings of single techniques. Scatterplots of different projections, for example, may be combined by coloring and linking subsets of points in all projections. In a similar fashion, linking and brushing can be applied to visualizations generated by all visualization techniques described above. As a result, the brushed points are highlighted in all visualizations, making it possible to detect dependencies and correlations. Interactive changes made in one visualization are automatically reflected in the other visualizations. Note that connecting multiple visualizations through interactive linking and brushing provides more information than considering the component visualizations independently.

Typical examples of visualization techniques which are combined by linking and brushing are multiple scatterplots, bar charts, parallel coordinates, pixel displays, and maps. Most interactive data exploration systems allow some form of linking and brushing. Examples are Polaris (see Fig. 7 in [8]) and the Scalable Framework (see Figs. 12 and 14 in [10]). Other tools and systems include S Plus [64], XGobi [38], [65], Xmdv [14], and DataDesk [66], [67].

6 Conclusion

The exploration of large data sets is an important but difficult problem. Information visualization techniques may help to solve the problem. Visual data exploration has high potential and many applications, such as fraud detection and data mining, will use information visualization technology for an improved data analysis.

Future work will involve the tight integration of visualization techniques with traditional techniques from such disciplines as statistics, machine learning, operations research, and simulation. Integration of visualization techniques and these more established methods would quickly combine automatic data mining algorithms with the intuitive power of the human mind, improving the quality and speed of the visual data mining process. Viusal data mining techniques also need to be tightly integrated with the systems used to manage the vast amounts of relational and semistructured information, including database management and data warehouse systems. The ultimate goal is to bring the power of visualization technology to every desktop to allow a better, faster, and more intuitive exploration of very large data resources. This will not only be valuable in an economic sense, but will also stimulate and delight the user.

Acknowledgments

This is an extended version of [6], portions of which are copyrighted by ACM.

References

[1] B. Shneiderman, "The Eye Have It: A Task by Data Type Taxonomy for Information Visualizations," *Visual Languages,* 1996.
[2] S. Card, J. Mackinlay, and B. Shneiderman, *Readings in Information Visualization.* Morgan Kaufmann, 1999.
[3] C. Ware, *Information Visualization: Perception for Design.* Morgan Kaufman, 2000.
[4] B. Spence, *Information Visualization.* Pearson Education Higher Education Publishers, 2000.
[5] H. Schumann and W. Müller, *Visualisierung: Grundlagen und allgemeine Methoden.* Springer, 2000.
[6] D. Keim, "Visual Exploration of Large Databases," *Comm. ACM,* vol. 44, no. 8, pp. 38-44, 2001.
[7] L. Nowell, S. Havre, B. Hetzler, and P. Whitney, "Themeriver: Visualizing Thematic Changes in Large Document Collections," *IEEE Trans. Visualization and Computer Graphics,* vol. 8, no. 1, pp. 9-20, Jan.-Mar. 2002.
[8] D. Tang, C. Stolte, and P. Hanrahan, "Polaris: A System for Query, Analysis and Visualization of Multidimensional Relational Databases," *IEEE Trans. Visualization and Computer Graphics,* vol. 8, no. 1, pp. 52-65, Jan.-Mar. 2002.
[9] J. Abello and J. Korn, "MGV: A System for Visualizing Massive Multidigraphs," *IEEE Trans. Visualization and Computer Graphics,* vol. 8, no. 1, pp. 21-38, Jan.-Mar. 2002.

[10] N. Lopez, M. Kreuseler, and H. Schumann, "A Scalable Framework for Information Visualization," *IEEE Trans. Visualization and Computer Graphics,* vol. 8, no. 1, pp. 39-51, Jan.-Mar. 2002.
[11] D. Keim, "Designing Pixel-Oriented Visualization Techniques: Theory and Applications," *IEEE Trans. Visualization and Computer Graphics,* vol. 6, no. 1, pp. 59-78, Jan.-Mar. 2000.
[12] B. Shneiderman, "Tree Visualization with Treemaps: A 2D Space-Filling Approach," *ACM Trans. Graphics,* vol. 11, no. 1, pp. 92-99, 1992.
[13] B. Johnson and B. Shneiderman, "Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information," *Proc. Visualization '91 Conf.,* pp. 284-291, 1991.
[14] M.O. Ward, "Xmdvtool: Integrating Multiple Methods for Visualizing Multivariate Data," *Proc. Visualization 94,* pp. 326-336, 1994.
[15] D. Asimov, "The Grand Tour: A Tool for Viewing Multidimensional Data," *SIAM J. Science & Statistical Computing,* vol. 6, pp. 128-143, 1985.
[16] A. Inselberg and B. Dimsdale, "Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry," *Proc. Visualization '90,* pp. 361-370, 1990.
[17] J.A. Wise, J.J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow, "Visualizing the Non-Visual: Spatial Analysis and Interaction with Information from Text Documents," *Proc. Symp. Information Visualization,* pp. 51-58, 1995.
[18] C. Chen, *Information Visualisation and Virtual Environments.* London: Springer-Verlag, 1999.
[19] M. Dodge, "Web Visualization," http://www.geog.ucl.ac.uk/casa/martin/geography_of_cyberspace.html, Oct. 2001.
[20] G.D. Battista, P. Eades, R. Tamassia, and I.G. Tollis, *Graph Drawing.* Prentice Hall, 1999.
[21] J. Trilk, "Software Visualization," http://wwwbroy.informatik.tu-muenchen.de/~trilk/sv.html, Oct. 2001.
[22] D.F. Andrews, "Plots of High-Dimensional Data," *Biometrics,* vol. 29, pp. 125-136, 1972.
[23] W.S. Cleveland, *Visualizing Data.* Summit, N.J.: Hobart Press, 1993.
[24] P.J. Huber, "The Annals of Statistics," *Projection Pursuit,* vol. 13, no. 2, pp. 435-474, 1985.
[25] G.W. Furnas and A. Buja, "Prosections Views: Dimensional Inference through Sections and Projections," *J. Computational and Graphical Statistics,* vol. 3, no. 4, pp. 323-353, 1994.
[26] R. Spence, L. Tweedie, H. Dawkes, and H. Su, "Visualization for Functional Design," *Proc. Int'l Symp. Information Visualization (InfoVis '95),* pp. 4-10, 1995.
[27] J.J. van Wijk and R.D. van Liere, "Hyperslice," *Proc. Visualization '93,* pp. 119-125, 1993.
[28] H. Chernoff, "The Use of Faces to Represent Points in k-Dimensional Space Graphically," *J Am. Statistical Assoc.,* vol. 68, pp. 361-368, 1973.
[29] R.M. Pickett and G.G. Grinstein, "Iconographic Displays for Visualizing Multidimensional Data," *Proc. IEEE Conf. Systems, Man, and Cybernetics,* pp. 514-519, 1988.
[30] H. Levkowitz, "Color Icons: Merging Color and Texture Perception for Integrated Visualization of Multiple Parameters," *Proc. Visualization '91,* pp. 22-25, 1991.
[31] D.A. Keim and H.-P. Kriegel, "Visdb: Database Exploration Using Multidimensional Visualization," *Computer Graphics & Applications,* vol. 6, pp. 40-49, Sept. 1994.
[32] M. Hearst, "Tilebars: Visualization of Term Distribution Information in Full Text Information Access," *Proc. ACM Human Factors in Computing Systems Conf. (CHI '95),* pp. 59-66, 1995.
[33] D.A. Keim, H.-P. Kriegel, and M. Ankerst, "Recursive Pattern: A Technique for Visualizing very Large Amounts of Data," *Proc. Visualization '95,* pp. 279-286, 1995.
[34] M. Ankerst, D.A. Keim, and H.-P. Kriegel, "Circle Segments: A Technique for Visually Exploring Large Multidimensional Data Sets," *Proc. Visualization '96, Hot Topic Session,* 1996.
[35] J. LeBlanc, M.O. Ward, and N. Wittels, "Exploring n-Dimensional Databases," *Proc. Visualization '90,* pp. 230-239, 1990.
[36] S. Feiner and C. Beshers, "Visualizing n-Dimensional Virtual Worlds with n-Vision," *Computer Graphics,* vol. 24, no. 2, pp. 37-38, 1990.
[37] G.G. Robertson, J.D. Mackinlay, and S.K. Card, "Cone Trees: Animated 3D Visualizations of Hierarchical Information," *Proc. Human Factors in Computing Systems CHI 91 Conf.,* pp. 189-194, 1991.
[38] D.F. Swayne, D. Cook, and A. Buja, "User's Manual for XGobi: A Dynamic Graphics Program for Data Analysis," Bellcore technical memorandum, 1992.
[39] A. Buja, D.F. Swayne, and D. Cook, "Interactive High-Dimensional Data Visualization," *J. Computational and Graphical Statistics,* vol. 5, no. 1, pp. 78-99, 1996.
[40] L. Tierney, *Lispstat: An Object-Orientated Environment for Statistical Computing and Dynamic Graphics.* New York: Wiley, 1991.
[41] D.B. Carr, E.J. Wegman, and Q. Luo, "Explorn: Design Considerations Past and Present," Technical Report No. 129, Center for Computational Statistics, George Mason Univ., 1996.
[42] E.A. Bier, M.C. Stone, K. Pier, W. Buxton, and T. DeRose, "Toolglass and Magic Lenses: The See-Through Interface," *Proc. SIGGRAPH '93,* pp. 73-80, 1993.
[43] K. Fishkin and M.C. Stone, "Enhanced Dynamic Queries via Movable Filters," *Proc. Human Factors in Computing Systems CHI '95 Conf.,* pp. 415-420, 1995.
[44] A. Spoerri, "Infocrystal: A Visual Tool for Information Retrieval," *Proc. Visualization '93,* pp. 150-157, 1993.
[45] C. Ahlberg and B. Shneiderman, "Visual Information Seeking: Tight coupling of Dynamic Query Filters with Starfield Displays," *Proc. Human Factors in Computing Systems CHI '94 Conf.,* pp. 313-317, 1994.
[46] S. G. Eick, "Data Visualization Sliders," *Proc. Seventh Ann. ACM Symp. User Interface Software and Technology (UIST),* pp. 119-120, 1994.
[47] J. Goldstein and S.F. Roth, "Using Aggregation and Dynamic Queries for Exploring Large Data Sets," *Proc. Human Factors in Computing Systems CHI '94 Conf.,* pp. 23-29, 1994.
[48] R. Rao and S.K. Card, "The Table Lens: Merging Graphical and Symbolic Representation in an Interactive Focus+Context Visualization for Tabular Information," *Proc. Human Factors in Computing Systems CHI 94 Conf.,* pp. 318-322, 1994.
[49] K. Perlin and D. Fox, "Pad: An Alternative Approach to the Computer Interface," *Proc. SIGGRAPH,* pp. 57-64, 1993.
[50] B. Bederson, "Pad++: Advances in Multiscale Interfaces," *Proc. Human Factors in Computing Systems CHI '94 Conf.,* p. 315, 1994.
[51] B.B. Bederson and J.D. Hollan, "Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics," *Proc. Seventh Ann. ACM Symp. User Interface Software and Technology (UIST),* pp. 17-26, 1994.
[52] C. Ahlberg and E. Wistrand, "Ivee: An Information Visualization and Exploration Environment," *Proc. Int'l Symp. Information Visualization,* pp. 66-73, 1995.
[53] V. Anupam, S. Dar, T. Leibfried, and E. Petajan, "Dataspace: 3D Visualization of Large Databases," *Proc. Int'l Symp. Information Visualization,* pp. 82-88, 1995.
[54] D. Schaffer, Z. Zuo, L. Bartram, J. Dill, S. Dubs, S. Greenberg, and M. Roseman, "Comparing Fisheye and Full-Zoom Techniques for Navigation of Hierarchically Clustered Networks," *Proc. Graphics Interface (GI '93),* pp. 87-96, 1993.
[55] Y. Leung and M. Apperley, "A Review and Taxonomy of Distortion-Oriented Presentation Techniques," *Proc. Human Factors in Computing Systems CHI '94 Conf.,* pp. 126-160, 1994.
[56] M.S.T. Carpendale, D.J. Cowperthwaite, and F.D. Fracchia, *IEEE Computer Graphics and Applications,* special issue on information visualization, vol. 17, no. 4, pp. 42-51, July 1997.
[57] R. Spence and M. Apperley, "Data Base Navigation: An Office Environment for the Professional," *Behaviour and Information Technology,* vol. 1, no. 1, pp. 43-54, 1982.
[58] J.D. Mackinlay, G.G. Robertson, and S.K. Card, "The Perspective Wall: Detail and Context Smoothly Integrated," *Proc. Human Factors in Computing Systems CHI '91 Conf.,* pp. 173-179, 1991.
[59] G. Furnas, "Generalized Fisheye Views," *Proc. Human Factors in Computing Systems CHI 86 Conf.,* pp. 18-23, 1986.
[60] M. Sarkar and M. Brown, "Graphical Fisheye Views," *Comm. ACM,* vol. 37, no. 12, pp. 73-84, 1994.
[61] J. Lamping, R. Rao, and P. Pirolli, "A Focus + Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies," *Proc. Human Factors in Computing Systems CHI 95 Conf.,* pp. 401-408, 1995.
[62] T. Munzner and P. Burchard, "Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space," *Proc. Virtual Reality Modeling Language (VRML '95) Symp.,* pp. 33-38, 1995.
[63] B. Alpern and L. Carter, "Hyperbox," *Proc. Visualization '91,* pp. 133-139, 1991.

[64] R. Becker, J.M. Chambers, and A.R. Wilks, *The New s Language.* Pacific Grove, Calif.: Wadsworth & Brooks/Cole Advanced Books and Software, 1988.

[65] R.A. Becker, W.S. Cleveland, and M.-J. Shyu, "The Visual Design and Control of Trellis Display," *J. Computational and Graphical Statistics,* vol. 5, no. 2, pp. 123-155, 1996.

[66] P.F Velleman, *Data Desk 4.2: Data Description.* Ithaca, N.Y.: Data Desk, 1992.

[67] A. Wilhelm, A.R. Unwin, and M. Theus, "Software for Interactive Statistical Graphics—A Review," *Proc. Int'l Softstat 95 Conf.,* 1995.

Daniel A. Keim received his diploma (equivalent to the MS degree) in computer science from the University of Dortmund in 1990 and the PhD degree in computer science from the University of Munich in 1994. He has been an assistant professor in the Computer Science Department of the University of Munich, an associate professor in the Computer Science Department of the Martin-Luther-University Halle, and a full professor in the Computer Science Department of the University of Constance. Currently, he is on leave from the University of Constance, working at AT&T Shannon Research Labs, Florham Park, New Jersey. He is working in the area of information visualization and data mining. In the field of information visualization, he developed several novel techniques which use visualization technology for the purpose of exploring large databases. He has published extensively on information visualization and data mining; he has given tutorials on related issues at several large conferences, including Visualization, SIGMOD, VLDB, and KDD; he has been program cochair of the IEEE Information Visualization Symposia in 1999 and 2000; he is program cochair of the ACM SIGKDD conference in 2002; and he is an editor of the *IEEE Transactions on Visualization and Computer Graphics* and the *Information Visualization Journal*. He is a member of the IEEE Computer Society.